SVM-COVID.R

karanYsingh

2021-04-05

```
# Loading data
knitr::opts_chunk$set(cache =TRUE)
covid_19_data = read.csv("C:\\Users\\91828\\Documents\\Rlab\\Covid\\Cleaned_Data.csv", header = TRUE)
data = covid_19_data[0:1000,]
colnames = c('Fever','Sore.Throat','Severity_Severe','Severity_Mild','Severity_Moderate')
data = data[colnames]
View(data)

set.seed(1)
# Load the data and remove NAs
data <- na.omit(data)

##Splitting data into test and train.
y <- c(rep(-1,length(data)/2), rep(1,length(data))/2)
data[y==1,] = data[y==1,] + 1

#Create our own test data
x <- data[1:2]
y <- data$Severity_Mild
y1 <- data$Severity_Moderate
y2 <- data$Severity_Severe
plot(x,col=(3-y))

library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.4
```

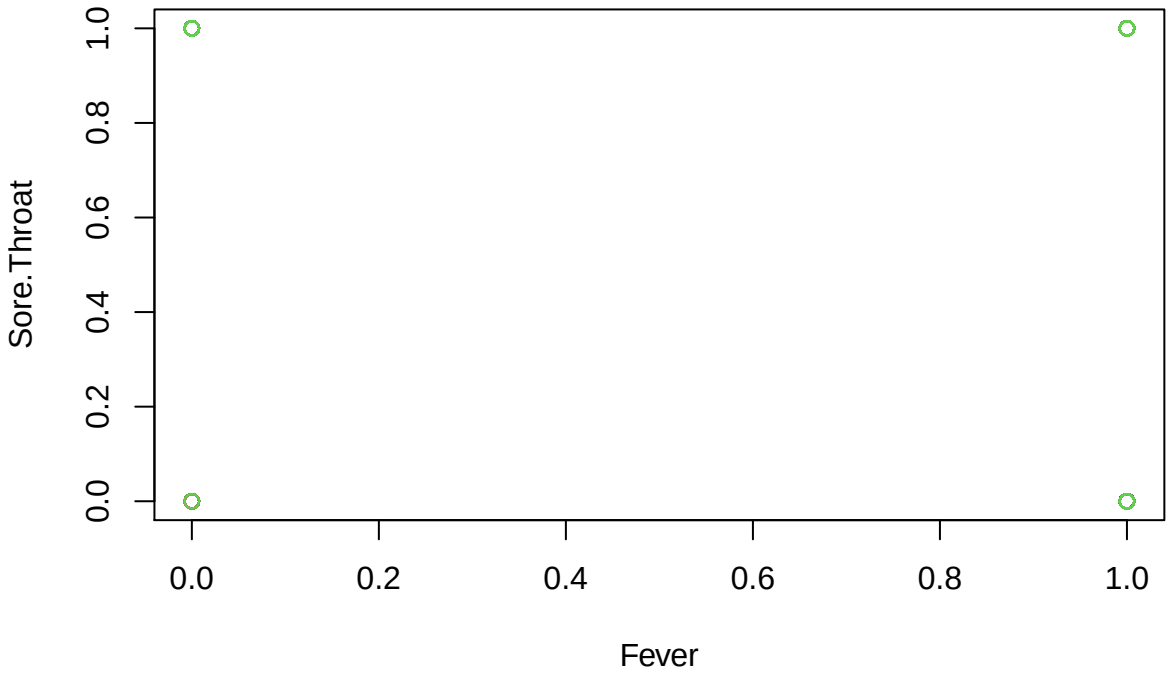
```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.3
```




```
dat <- data.frame(x=x, y=as.factor(y))
dat2 <- data.frame(x=x,y=as.factor(y1))
dat3 <- data.frame(x=x,y=as.factor(y2))
train <- sample(2 * length(y), length(y))

########################
set.seed(1)
tune <- tune(svm, y ~., data=dat[train,],
                 kernel='radial',
                 ranges = list(cost=c(0.1,1,10,100,1000),
                               gamma=c(0.5, 1,2,3,4)))

set.seed(1)
tune2 <- tune(svm, y ~., data=dat2[train,],
                 kernel='radial',
                 ranges = list(cost=c(0.1,1,10,100,1000),
                               gamma=c(0.5, 1,2,3,4)))


set.seed(1)
tune3 <- tune(svm, y ~., data=dat3[train,],
                 kernel='radial',
                 ranges = list(cost=c(0.1,1,10,100,1000),
                               gamma=c(0.5, 1,2,3,4)))

set.seed(1)
```

```
tune4 <- tune(svm, y ~., data=dat[train,],
              kernel='linear',
              ranges = list(cost=c(0.1,1,10,100,1000),
                            gamma=c(0.5, 1,2,3,4)))

summary(tune$best.model)
```

```
##
## Call:
## best.tune(method = svm, train.x = y ~ ., data = dat[train, ], ranges = list(cost = c(0.1,
##     1, 10, 100, 1000), gamma = c(0.5, 1, 2, 3, 4)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  0.1
##
## Number of Support Vectors:  246
##
##  ( 123 123 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```
summary(tune2$best.model)
```

```
##
## Call:
## best.tune(method = svm, train.x = y ~ ., data = dat2[train, ], ranges = list(cost = c(0.1,
##     1, 10, 100, 1000), gamma = c(0.5, 1, 2, 3, 4)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  0.1
##
## Number of Support Vectors:  258
##
##  ( 129 129 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```r
summary(tune3$best.model)
```

```
##
## Call:
## best.tune(method = svm, train.x = y ~ ., data = dat3[train, ], ranges = list(cost = c(0.1,
##     1, 10, 100, 1000), gamma = c(0.5, 1, 2, 3, 4)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  0.1
##
## Number of Support Vectors:  268
##
##  ( 134 134 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
```

```r
yhat <- predict(tune$best.model, dat[-train,])
yhat2 <- predict(tune2$best.model, dat2[-train,])
yhat3 <- predict(tune3$best.model, dat3[-train,])
yhat4 <- predict(tune4$best.model, dat3[-train,])

confusionMatrix(yhat, dat[-train,'y'])#radial Mild Severity
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 373 129
##          1   0   0
##
##                Accuracy : 0.743
##                  95% CI : (0.7024, 0.7807)
##     No Information Rate : 0.743
##     P-Value [Acc > NIR] : 0.5237
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.000
##             Specificity : 0.000
##          Pos Pred Value : 0.743
##          Neg Pred Value :   NaN
##              Prevalence : 0.743
##          Detection Rate : 0.743
```

```
##    Detection Prevalence : 1.000
##       Balanced Accuracy : 0.500
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(yhat2, dat2[-train,'y'])#radial Moderate Severity
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 381 121
##          1   0   0
##
##                Accuracy : 0.759
##                  95% CI : (0.7191, 0.7958)
##     No Information Rate : 0.759
##     P-Value [Acc > NIR] : 0.5244
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.000
##             Specificity : 0.000
##          Pos Pred Value : 0.759
##          Neg Pred Value :   NaN
##              Prevalence : 0.759
##          Detection Rate : 0.759
##    Detection Prevalence : 1.000
##       Balanced Accuracy : 0.500
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(yhat3, dat3[-train,'y'])#radial Severe Severity
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 387 115
##          1   0   0
##
##                Accuracy : 0.7709
##                  95% CI : (0.7316, 0.807)
##     No Information Rate : 0.7709
##     P-Value [Acc > NIR] : 0.525
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
```

```
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.7709
##          Neg Pred Value :    NaN
##              Prevalence : 0.7709
##          Detection Rate : 0.7709
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

```
confusionMatrix(yhat4, dat3[-train,'y'])#linear Sever Severity
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 387 115
##          1   0   0
##
##                Accuracy : 0.7709
##                  95% CI : (0.7316, 0.807)
##     No Information Rate : 0.7709
##     P-Value [Acc > NIR] : 0.525
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.7709
##          Neg Pred Value :    NaN
##              Prevalence : 0.7709
##          Detection Rate : 0.7709
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```